JavaOne℠

Sun's Worldwide Java Developer Conference

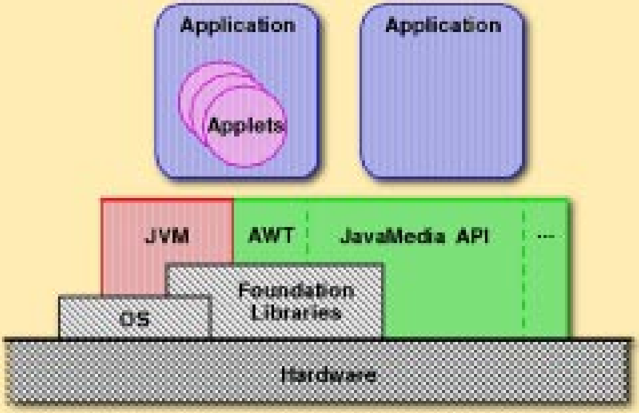
JavaOne℠

Sun's Worldwide Java Developer Conference

Java™ Media – The Java Multimedia APIs

J. Duane Northcutt

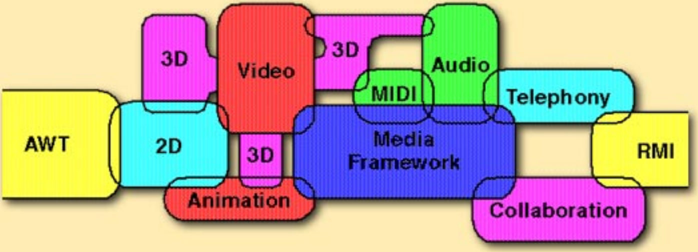
Java Platform Architecture



Java Media API Areas

- Java 2D – 2D Graphics and Imaging
- Java Media Framework – Time-Critical Media
    - Audio
    - Video
    - MIDI
- Java Animation – 2D Object Animation
- Java Telephony – Computer Telephony Integration
- Java Share – Interactive Multi-User Application and Data Sharing
- Java 3D – 3D Graphics and Behavior

Java Media API Area Relationships



Java Media API Areas

- **Java 2D – 2D Graphics and Imaging**
- Java Media Framework – Time-Critical Media
- Java Animation – 2D Object Animation
- Java Telephony – Computer Telephony Integration
- Java Share – Interactive Multi-User Application and Data Sharing
- Java 3D – 3D Graphics and Behavior

Java 2D – 2D Graphics and Imaging

Java 2D – 2D Graphics and Imaging – Introduction

- High quality device-independent graphics
  - Augmented line art/shape drawing
  - Enhanced fonts, rich text
  - Added in-memory images to streaming model
- Single comprehensive rendering model
  - Spatial transformations (e.g., rotate, scale, skew)
  - Flexible compositing/blending mechanisms
  - Device (in)dependent color and coordinate spaces

Java 2D – 2D Graphics and Imaging – Key Features

- Extensible via subclassing and loadable content handlers
  - Displays/printing devices
  - Image i/o formats
  - Image processing/transformation filters
- Designed to be integrated with AWT
  - Provide new features (e.g., buffered images)
  - Augment existing AWT classes (e.g., graphics, fonts)
  - Introduce new classes to broaden AWT (e.g., color spaces)
- High-performance rendering path to display area

Java 2D – 2D Graphics and Imaging – Interface Overview

- Graphic – abstract drawing target; encapsulates state
  - Drawing methods; clip region, transformations, compositing, font
- Path – collection of points defining outline of shape
  - Lines/bezier curves/gliph; can be stroked or filled
- Font – more control over font characteristics/ information
  - Support various font technologies, allow custom fonts
- Image – buffered in-memory; defined format and color space
  - Transformations, compositing, image filters

Java 2D – 2D Graphics and Imaging – Interface Overview

- Paint – more sophisticated means of filling a shape
  - Gradients/patterns/etc.; alternative to simple color
- Stroke – boundary of a shape
  - Width, join/cap styles, color/paint
- Color – translations between device (in)dependent spaces
  - Defaults for common cases; to/from rgb709/CIE XYZ;ICC profiles

Java 2D – 2D Graphics and Imaging – Interface Overview

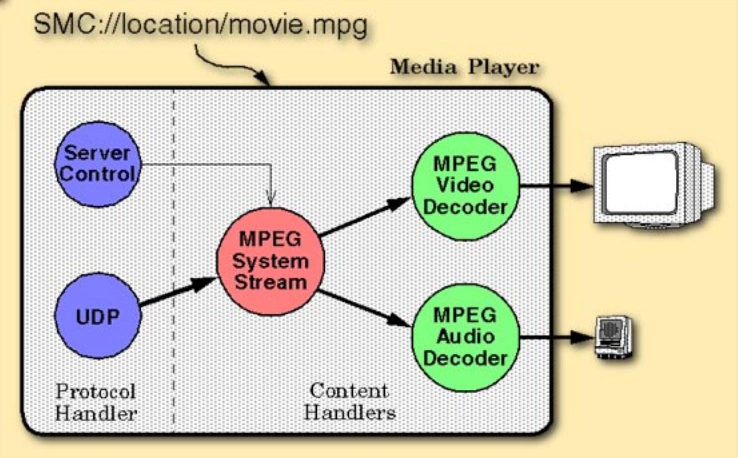
- Transformation – set of transformations on points/paths
  - Affine (translate, rotate, skew, etc.) or custom transforms
- Composite – definition of how colors are combined
  - Overlays, blending, transparency, etc.
- Image filter – imaging operations, built-in/ user-extensible
  - Table lookup, convolve, sharpen

Java 2D – 2D Graphics and Imaging

Java Media API Areas

- Java 2D – 2D Graphics and Imaging
- **Java Media Framework – Time-Critical Media**
- Java Animation – 2D Object Animation
- Java Telephony – Computer Telephony Integration
- Java Share – Interactive Multi-User Application and Data Sharing
- Java 3D – 3D Graphics and Behavior

Java Media Framework



Java Media Framework – Introduction

- Media Player – high-level, universal, media presentation tool
  - Simple interface based on defined media naming scheme (URI)
  - Dynamic composition (over the net) from interoperable modules
  - Demultiplexing and synchronization of multi-channel content
  - Interfaces for control and coordination by/with other entities
- Media Framework – connection graph architecture
  - Permits creation of arbitrary real-time processing pipelines
  - Synchronizes time-critical activities through logical time systems
  - Provides latency management mechanisms (pre-fetch/pre-roll)

Java Media Framework – Key Features

- Media Players – uniform encapsulation of media handlers
  - Extensible transport protocols, media types and coding formats
  - Mix-and-match and reuse of framework objects
  - Transparent use of native methods where necessary
- Connection graphs – dynamic, hierarchical, composition
  - Use player alone to present timed media
  - Synchronize players with other players/other activities
  - Add functions to stream processing pipes
  - Create new protocol/content handler modules

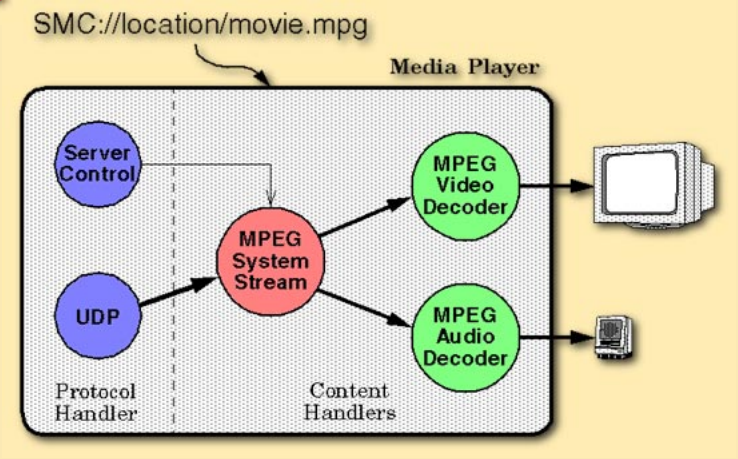
Java Media Framework – Interface Overview

- Manager – high-level, media-specific, player control
  - Media player generation, connection graph editing
- Node – basic building block of timed media processing graph
  - I/O connections, time/latency control interfaces, format objects
- Connection – front-specific data paths between nodes
  - Type/format negotiation, flow control, buffer management
- Clock – basic entity in construction of logical time systems
  - Start/stop/scale relative to a timebase

Java Media Framework – Media Specific Components

- Video – defines basic data formats and control interfaces
  - Supports both latency sensitive/insensitive video
  - Accommodates both steaming and stored video sources
- Audio – defines basic data formats and control interfaces
  - Supports sampled, synthesized audio, 3D/spatial audio source
  - Accommodates both streaming and stored audio sources
- MIDI – defines interfaces for extension and integration
  - Unique synchronization requirements – timed event streams
  - Loadable, synthesizers, effects, etc.

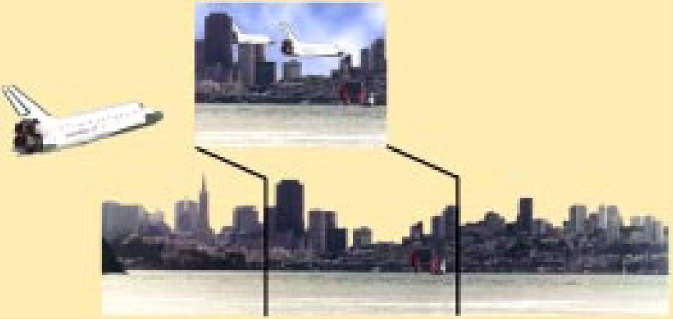
Java Media Framework



Java Media API Areas

- Java 2D – 2D Graphics and Imaging
- Java Media Framework – Time-Critical Media
- **Java Animation – 2D Object Animation**
- Java Telephony – Computer Telephony Integration
- Java Share – Interactive Multi-User Application and Data Sharing
- Java 3D – 3D Graphics and Behavior

Java Animation – 2D Object Animation



Java Animation – 2D Object Animation – Introduction

- High-performance 2D animation
  - Multiple (fixed and sequenced) sprites with transparency
  - Programmed sprite behaviors
  - Scrolling background images
  - Aggregation/hierarchical composition
  - Display order control and mixing
  - Image transformation effects
  - Sprite collision detection
  - Integrated with AWT, Java 2D, Java 3D, and Media Framework

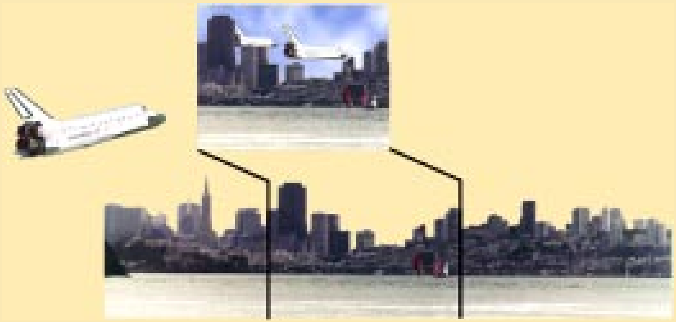
Java Animation – 2D Object Animation – Key Features

- Displayable object attributes
  - Position – location in three-space relative to destination surface
  - Visibility – currently in hidden or visible state
  - Viewport – displayable (rectangular) region of image
  - Image source – data that defines the object’s image
  - Destination surface – logical rendering target of image
- Primary animation methods
  - get/set/adjust object attributes
  - add/apply/prepare/enumerate effects modules
  - get/set/ test for object collisions

Java Animation – 2D Object Animation – Interface Overview

- Displayable objects – dynamically changeable attributes
  - Backgrounds, sprites, group, surface
- Source images – image data from variety of sources
  - Single/sequence of images; from Java 2D/Media Framework
- Real-time image effects – loadable image transformations
  - Applied on-the-fly; can be concatenated; e.g., flip/ shear, affine
- Collision detection facility – object with object or point
  - Collision defined by point or vector of rectangles

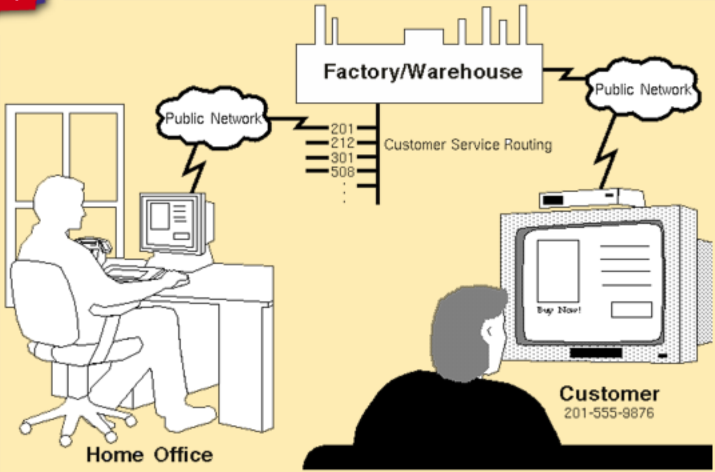
Java Animation – 2D Object Animation



Java Media API Areas

- Java 2D – 2D Graphics and Imaging
- Java Media Framework – Time-Critical Media
- Java Animation – 2D Object Animation
- **Java Telephony – Computer Telephony Integration**
- Java Share – Interactive Multi-User Application and Data Sharing
- Java 3D – 3D Graphics and Behavior

Java Telephony – Computer Telephony Integration



Java Telephony – Computer Telephony Integration – Introduction

- Telephony framework – high level interface to call control
  - Control-oriented – media framework can source/sink data
  - First-party – make/receive calls; desktops/PDAs/cell phones
  - Third-party – also automated call distribution/hand-off

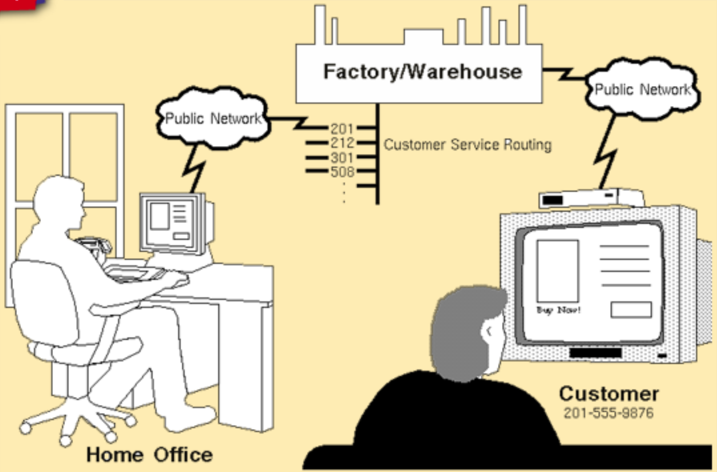
Java Telephony – Computer Telephony Integration – Key Features

- Flexible and powerful interface to telephony subsystem
  - Independent of network technology – POTS/ISDN/ ATM/Ethenet
  - Implementable on all call control engines – TSAPI/ TAPI/SunXTL
  - Transparent access to local or remote call engines/ protocol stacks
- Simple and intuitive interface
  - Small set of objects representing all key telephony entities
  - Observer/observable model for asynchronous event handling

Java Telephony – Computer Telephony Integration – Interface Overview

- Provider – interface to a particular telephony subsystem
  - Specified by URI; supports zero or more calls
- Call – representation of an instance of a telephone call
  - Contains zero or more communications links (i.e., connections)
- Connection – control relationship between a Device/Call pair
  - Encapsulates state of associated data stream(s)
- Device – addressable endpoint, access to telephony services
  - Logical entity, multiple devices per physical device possible

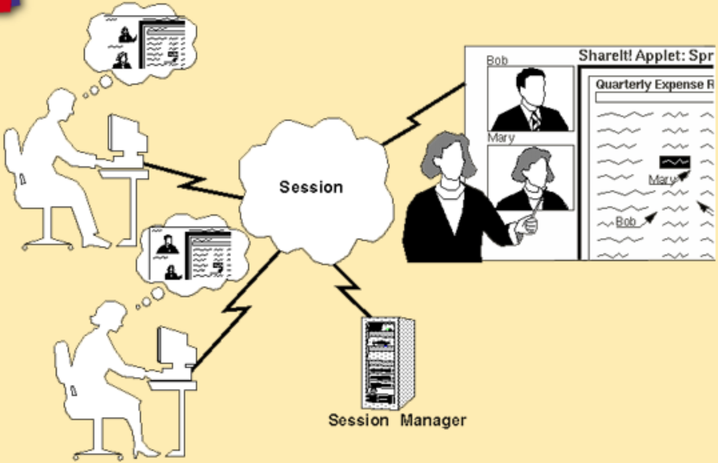
Java Telephony – Computer Telephony Integration



Java Media API Areas

- Java 2D – 2D Graphics and Imaging
- Java Media Framework – Time-Critical Media
- Java Animation – 2D Object Animation
- Java Telephony – Computer Telephony Integration
- **Java Share – Interactive Multi-User Application and Data Sharing**
- Java 3D – 3D Graphics and Behavior

Java Share – Collaboration and Data Sharing



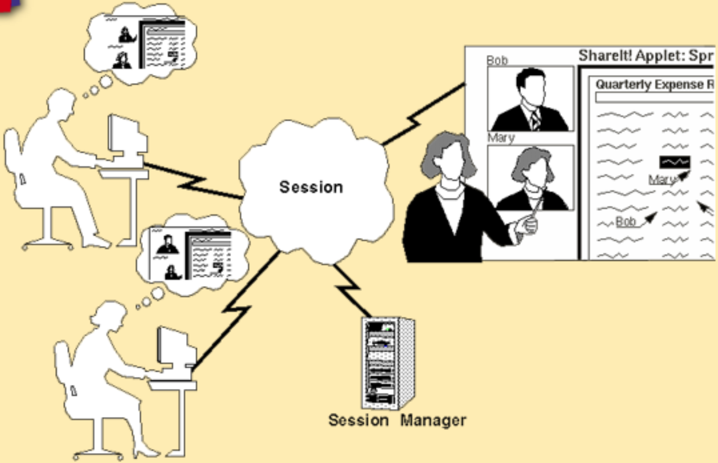
Java Share – Collaboration and Data Sharing – Introduction

- Real-time, interactive, multi-user sharing of applications
  - Cross-platform sharing of locally unavailable applications
  - Unanticipated sharing of *collaboration-unaware* applets
  - Anticipated sharing of *collaboration-aware* applets

Java Share – Collaboration and Data Sharing – Interface Overview

- Framework for sharing of collaboration-unaware applets
  - Either replicated or centralized architectural approach
  - Selection of range of floor control and serialization policies
  - Input event and graphics call distribution and (de)multiplexing
  - Meaning default selections
- Mechanisms for constructing collaboration-aware applets
  - Distributed shared objects
  - Session management services – Session
  - Multi-party communications facilities – Channel
  - Distributed synchronization mechanisms – Token
  - Choice of implementation protocols

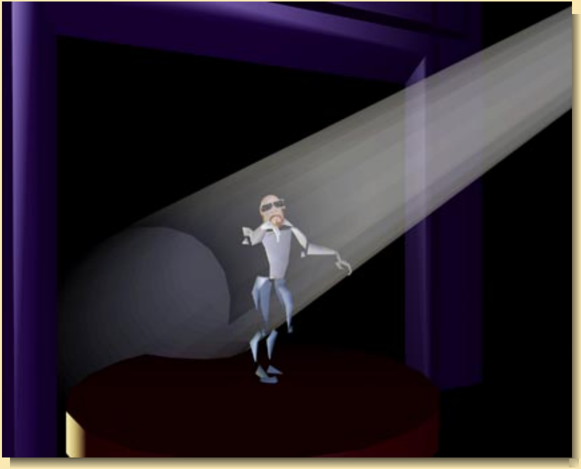
Java Share – Collaboration and Data Sharing



Java Media API Areas

- Java 2D – 2D Graphics and Imaging
- Java Media Framework – Time-Critical Media
- Java Animation – 2D Object Animation
- Java Telephony – Computer Telephony Integration
- Java Share – Interactive Multi-User Application and Data Sharing
- **Java 3D – 3D Graphics and Behavior**

Java 3D – 3D Geometry and Behavior



Java 3D – 3D Geometry and Behavior – Introduction

- High-performance, interactive, 3D graphics support
  - Immediate, retained and compiled-retained 3D graphics
  - High-level specification of behavior and control of 3D objects
- Simplifies 3D application programming
  - Higher level of abstraction than OpenGL/XGL
  - Also provides access to lower-level interfaces for performance
- Advanced, performance-oriented features
  - Generalized morphing engine
  - Optimized geometry and behavior compiler
  - High-resolution coordinate anchors

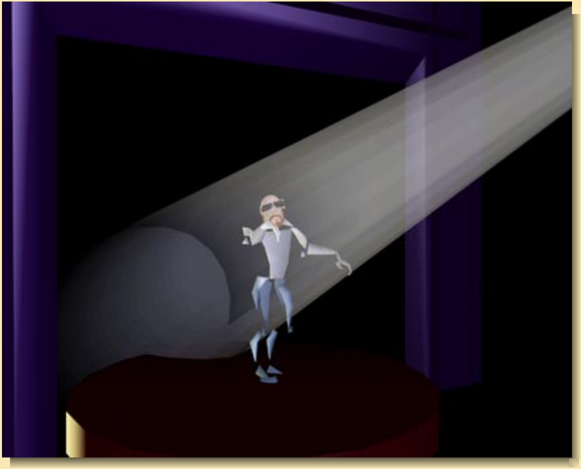
Java 3D – 3D Geometry and Behavior – Key Features

- Supports wide range of applications
  - Simple 3D objects on web pages
  - 3D browsers and authoring tools
  - 3D file format loaders and viewers (e.g., VRML)
  - Large-scale interactive virtual worlds
- Sophisticated view models – fishtank VR, HMD, portal/cave
  - Supports tracker-based viewpoints
- Closely integrated with other Java Media areas
  - Media Framework, Audio, Video, MIDI, and 2D Animation

Java 3D – 3D Geometry and Behavior – Interface Overview

- Hierarchical scene-graph construction – Group/ Leaf nodes
  - Spatial organization, appearance inheritance, hi-res references
- Animation – spline-based morphing operator
  - Object attribute morphing – e.g. color, orientation, shape, etc.
- Events – sensing nodes linking events to observers
  - Generation, propagation, and execution triggering
- Enables culling optimizations – rendering and behavior
  - Location-based, view frustum and occlusion culling

Java 3D – 3D Geometry and Behavior



Java Media API Release Plans and Status

- Progression of review cycles
  - Licensees of Java™-based technology
  - Early adopters (selected active developers)
  - Open review
- Java Media areas released as available
- Announcements through the JavaSoft web site
- Work currently underway on:
  - API specifications
  - Prototype implementations
  - Example applications/applets

Java Media API Roadmap

	Draft API Spec	Early Access	FCS
Java 2D	2Q96	2H96	1997
Framework	2Q96	2H96	1997
Java Animation	3Q96	2H96	1997
Java Share	2Q96	2H96	1997
Java Telephony	2Q96	2H96	1997
Java 3D	4Q96	1H96	1997

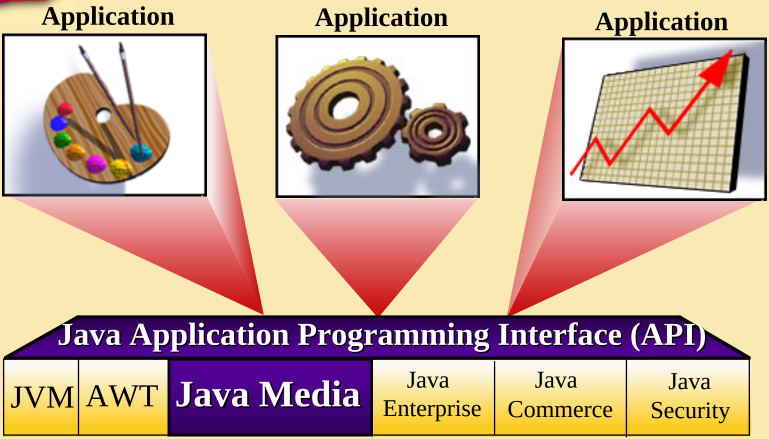
JavaOne℠

Sun's Worldwide Java Developer Conference

Java™ Media, The Java Media APIs

Eileen McGinnis

Java Media Basic Architecture



Application

Application

Application

Java Application Programming Interface (API)

JVM

AWT

Java Media

Java Enterprise

Java Commerce

Java Security

Java Media Contributors to Date

- Adobe
- Intel
- Lucent
- Macromedia
- SGI
- Sun

<u>Recent Additions</u>

- Apple
- Netscape

Framework for Composition & Synchronization

- Rendering–2D, 3D ~~*Time*~~
- Animation–2D, 3D with behaviors
- Streaming and Stored Media–Audio, Video, Midi
- Collaboration
- Computer Telephony Integration

Java Media Does Include

- Smooth integration of Media Objects—developed within the philosophy used to create Java™-based products
- Extensibility (e.g., new media, transports, data formats)
- Ability to supply or use native methods
- Interoperability among diverse media environments
- Your favorite (*insert media package here*)

Java Media Does NOT Include

- A winner chosen by Java Media for...
  - Formats
  - Codecs
  - Protocols
- A required distribution procedure with delivery cons
- Lowest common denominator effects

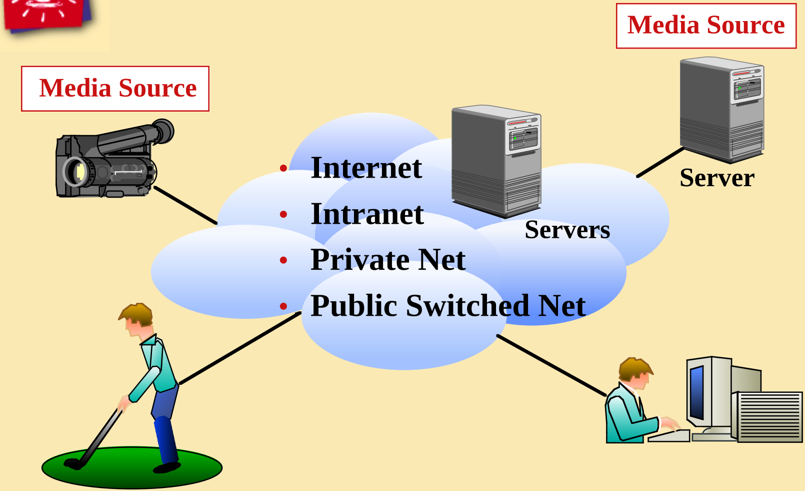
Java Media Enables

- New flexible application distribution model
- Innovation in higher levels
- Diverse value adds
- Competition/contribution at all levels

Java Media Is Designed for Tomorrow

	Media Source		*Destination*
Yesterday	Your Desk	→	Your Desk
	External Storage on Network	→	Your Desk
Today	Live Source	→	Your Desk
Tomorrow	Live Source	→	Wherever **You** Are

Java Media Is Designed for Tomorrow



Media Source

Media Source

- Internet
- Intranet
- Private Net
- Public Switched Net

Servers

Server